AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application:

LISTING OF CLAIMS:

1 - 75 canceled

76. (previously presented): The method of claim 81, wherein

a flip-flop is marked, if any of its corresponding latch interface constraints belong to the said unsatisfiable core

a flip-flop is marked, if its initial state value constraint belongs to the said unsatisfiable core.

77. (previously presented): The method of claim 81, wherein

a flip-flop is marked, if any of its corresponding latch interface constraints belong to the said unsatisfiable core

a flip-flop is not marked if only its initial state value belongs to the said unsatisfiable core, in which case a constraint for this initial input is added to the abstract model without adding the flip-flop.

78. (previously presented): The method of claim 81, wherein

a lazy constraint is used instead of an eager 1-literal constraint denoting an initial value of a flip-flop,

an initial value constraint m is replaced by (m+y)(m+!y), where y is introduced as a dummy variable to delay propagation of effect of said initial value constraint in the satisfiability solver,

said lazy constraint making it less likely that a flip-flop is marked due to its initial value constraint being present in the said unsatisfiable core, thereby leading to a smaller abstract model.

79. (currently amended): The method of claim 81, wherein

a lazy constraint is used instead of an eager 1-literal constraint denoting an environmental constraint, wherein an environmental constraint ~~such as~~ (m) is replaced by (m+y)(m+!y), where y is introduced as a dummy variable to delay the propagation of the effect of the said environmental constraint in the satisfiability solver.

the said lazy constraint making it less likely that an external constraint node is marked due to its environmental constraint being present in the said unsatisfiable core, thereby leading to a smaller abstract model ~~in many cases~~.

80. (canceled):

81. (currently amended): A computer implemented method for generating an abstract model for a sequential design of an electronic circuit for verification of a given correctness property, comprising the steps of:

(a) unrolling the sequential design of the electronic circuit with the given correctness property, and environmental constraints up to some depth *k*

(b) solving the resulting satisfiability problem to determine whether the ~~said~~ given correctness property is violated

(c) deriving an unsatisfiable core from the proof of unsatisfiability when the given correctness property is not violated, where an unsatisfiable core is a subset of the constraints that is guaranteed to be sufficient for showing that the problem is unsatisfiable

(d) using the said unsatisfiable core to derive an abstract model of the sequential design of the electronic circuit for further verification of the sequential design, wherein,

a subset of flip-flops and external constraint nodes in the sequential design of the electronic circuit are marked based on certain related constraints being present in said unsatisfiable core

the said abstract model consisting of complete combinational fanin cones of only the marked flip-flops and the marked external constraint nodes, such that outputs from the unmarked flip-flops are regarded as pseudo-primary inputs, and

said abstract model providing a benefit that it is guaranteed to preserve the correctness of the ~~said~~ given correctness property up to the said finite depth *k*.

82. (currently amended): A method for generating an abstract model for a sequential design of an electronic circuit for verification of a given correctness property, comprising:

a) unrolling the sequential design of the electronic circuit to depth *k* (where $k = kmin$ at the start)

b) deriving an abstract model of the sequential design of the electronic circuit at depth *k*, that preserves correctness of the given correctness ~~said~~ property up to depth *k*, when the given correctness property is not violated

c) iteratively increasing *k* up to some limit *kmax*, and repeating above steps (a - c) until either the size of the said abstract model of the sequential design of the electronic circuit remains ~~stable~~ unchanged over a ~~certain~~ predetermined number of consecutive depths, or the limit *kmax* is reached.

83. (Previously presented): The method of Claim 82, where the size of the abstract model corresponds to the number of flip-flops in the abstract model.

84. (Previously presented): The method of Claim 82, where the method of Claim 82 is used to derive the abstract model at depth *k*.